

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VICENTE COSTAMILAN DA CUNHA

**Métodos de Segmentação Automática de Sinais
de Eletromiografia de Superfície para
Classificação de Movimentos Utilizando Redes
Neurais Artificiais**

Porto Alegre

2015

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VICENTE COSTAMILAN DA CUNHA

**Métodos de Segmentação Automática de Sinais de
Eletromiografia de Superfície para Classificação de
Movimentos Utilizando Redes Neurais Artificiais**

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Eng. Alexandre Balbinot

Porto Alegre

2015

INSERIR A FICHA BIBLIOGRÁFICA

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
GRADUAÇÃO EM ENGENHARIA ELÉTRICA

VICENTE COSTAMILAN DA CUNHA

**Métodos de Segmentação Automática de Sinais de
Eletromiografia de Superfície para Classificação de
Movimentos Utilizando Redes Neurais Artificiais**

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica

Trabalho aprovado. Porto Alegre, XX de XXXXXX de 2015:

Prof. Dr. Eng. Alexandre Balbinot
Orientador

Professor
Convidado 1

Professor
Convidado 2

Porto Alegre
2015

*A Gilberto, meu pai, torre de razão.
Acaso lo que digo no es verdadero, ojalá sea profético.*

Agradecimentos

INSERIR AGRADECIMENTOS

*Take nothing on its looks;
take everything on evidence.
There's no better rule.*

Charles Dickens, Great Expectations

Resumo

A segmentação de sinais de eletromiografia (EMG) é parte essencial de pré-processamento em aplicações de reconhecimento de movimentos e controle de próteses, separando trechos de interesse do sinal correspondentes a esforços musculares e descartando trechos de sinal com baixa atividade muscular. Neste estudo, quatro métodos para segmentação automática de sinais de EMG foram implementados em MATLAB. Os métodos foram aplicados aos sinais de EMG de superfície a base de dados do projeto Ninapro e aos sinais da base de dados adquiridos pelo Laboratório de Instrumentação Eletro-Eletrônica da UFRGS. Uma rede neural artificial foi utilizada para classificar os movimentos realizados correspondentes aos sinais segmentados com os quatro métodos. TODO: RESULTADOS.

Palavras-chave: Eletromiografia. Segmentação. Base de dados Ninapro.

Abstract

TRADUZIR RESUMO

Keywords: Eletromiography. Segmentation. MATLAB. Ninapro database.

Lista de ilustrações

Figura 1 – Soma de potenciais de ação das n fibras de uma unidade motora, formando uma MUAP $h(t)$	16
Figura 2 – MUAPTs de diferentes MUs somam-se para compor o sinal adquirido por um canal de EMG.	17
Figura 3 – Fluxograma representativo do MTD1.	19
Figura 4 – Fluxograma representativo do MTD2.	20
Figura 5 – Fluxograma representativo do MTD3.	22
Figura 6 – Fluxograma representativo do MTD4.	24

Lista de tabelas

Tabela 1	–	Parâmetros utilizados para definir MTD1.	18
Tabela 2	–	Parâmetros utilizados para definir MTD2.	20
Tabela 3	–	Parâmetros utilizados para definir MTD3.	21
Tabela 4	–	Parâmetros utilizados para definir MTD4.	23

Lista de abreviaturas e siglas

EMG	Eletromiografia
MU	<i>Motor Unit</i>
MUAP	<i>Motor Unit Action Potencial</i>
MUAPT	<i>Motor Unit Action Potencial Trains</i>
MTD#	<i>Método número #</i>
BEP	<i>Beginning Extraction Point</i>
EEP	<i>Ending Extraction Point</i>

Sumário

1	INTRODUÇÃO	14
2	REVISÃO BIBLIOGRÁFICA	16
2.1	Sinais de Eletromiografia	16
2.2	Métodos de Segmentação	17
2.2.1	Método 1 - método iterativo utilizando <i>thresholding</i> para detecção de centros de segmentos de comprimento constante (MTD1)	18
2.2.2	Método 2 - método não iterativo utilizando <i>thresholding</i> para detecção de centros de segmentos de comprimento constante (MTD2)	19
2.2.3	Método 3 - método com janela deslizante para detecção de BEP e EEP de segmentos utilizando variação total (MTD3)	21
2.2.4	Método 4 - método com janela deslizante para detecção de BEP e EEP de segmentos utilizando <i>thresholding</i> (MTD4)	23
2.3	Base de Dados Utilizadas	25
2.3.1	Base de Dados Ninapro	25
2.3.2	Base de Dados do IEE	25
3	METODOLOGIA EXPERIMENTAL	26
3.1	Preprocessamento	26
3.1.1	Retificação de Sinal	26
3.1.2	Normalização	26
3.1.3	MTD1 e MTD2: uso da função <i>findpeaks</i> para detecção de centros de segmentos	26
3.1.4	Obtenção de valores para parâmetros dos métodos	26
3.2	Rede Neural Artificial	26
3.2.1	Características dos segmentos utilizadas pela rede	26
3.2.2	Treinamento, Validação e Teste	26
4	CAPÍTULO TEMPORÁRIO: ORIENTAÇÃO	27
5	RESULTADOS E DISCUSSÕES	28
6	CONCLUSÕES	29
7	PROPOSTAS DE FUTUROS TRABALHOS	30

APÊNDICES	31
APÊNDICE A – FUNÇÃO EM MATLAB PARA MTD1	32
APÊNDICE B – FUNÇÃO EM MATLAB PARA MTD2	35
APÊNDICE C – FUNÇÃO EM MATLAB PARA MTD3	38
APÊNDICE D – FUNÇÃO EM MATLAB PARA MTD4	40

1 Introdução

Sinais de EMG apresentam crescentes aplicações no controle de próteses mioelétricas. Por exemplo, (HARGROVE *et al.* 2013) mostram o controle de uma prótese de perna de um amputado acima do joelho direito, enquanto (JUN-UK CHU *et al.* 2007) apresentou bons resultados de reconhecimento de padrões de EMG para desenvolvimento de uma prótese multifuncional de mão. Em áreas não relacionadas à próteses, (CONSTANTINOS S. PATTICHIS *et al.* 1995) utilizaram redes neurais artificiais para realização de diagnósticos clínicos de desordens neuromusculares.

As principais estratégias para caracterização de sinais de EMG e potenciais de ação das unidades motoras baseiam-se no uso de um método classificador. Métodos de classificação utilizados incluem - entre inúmeros outros - redes neurais artificiais (HUDGINS *et al.* 1993), classificador Bayesiano (ENGLEHART & HUDGINS, 2003) e lógica *fuzzy* (CHAN *et al.* 2000). Tais sistemas de classificação necessitam, como parte do pré-processamento, segmentar os sinais de EMG adquiridos, para então realizar extração de características dos segmentos como amplitude, número de cruzamentos por zero, coeficientes de autoregressão, transformadas de Fourier e, mais recentemente, transformadas Wavelet (JUN-UK CHU *et al.* 2007).

Este trabalho tem como objetivo desenvolver em MATLAB quatro diferentes métodos de segmentação automática de sinais de EMG. Os métodos serão aplicados em sinais da base de dados do projeto Ninapro (ATZORI *et al.* 2012) - que contém sinais de dez canais de EMG de superfície para 52 diferentes movimentos de mão e punho - e em sinais da base de dados similarmente adquiridos pelo Laboratório de Instrumentação Eletro-Eletrônica.

Os primeiros dois métodos (que serão identificados neste estudo pelos mnemônicos MTD1 e MTD2) tratam da detecção de picos do sinal utilizando *thresholding* e produzem segmentos de comprimento constante centrados nestes picos. O terceiro (MTD3) e quarto (MTD4) métodos utilizam de janela deslizante para identificação de pontos iniciais e finais dos segmentos, produzindo segmentos de comprimento variável. Para as implementações dos métodos neste estudo, assume-se o conhecimento *a priori* do número de segmentos de interesse contidos no sinal.

O primeiro método (MTD1) é baseado no método de segmentação utilizado em (CHAUVET *et al.* 2001). Trata-se de método iterativo, identificando os picos do sinal a partir de *threshold* de amplitude, segmentando o sinal em janelas de comprimento constante centradas nos picos. O valor de *threshold* para a primeira iteração corresponde ao máximo

absoluto do sinal. A cada nova iteração em que não se atinge um número desejado de segmentos, o novo *threshold* é calculado como fração do *threshold* da iteração anterior.

O segundo método (MTD2) é baseado no método de segmentação utilizado em (KATSIS *et al.* 2006). De forma similar ao MTD1, também utiliza *threshold* para detecção de picos do sinal e segmentação com janelas de comprimento constante em torno dos picos. Diferentemente de MTD1, MTD2 não é iterativo, utilizando o valor máximo e o comprimento do sinal para cálculo do valor de *threshold*.

O terceiro método (MTD3) é baseado no método de segmentação utilizado em (GUT & MOSCHYTZ, 2000). Uma janela deslizante percorre o sinal e identifica inícios de segmentos quando a declividade média no interior da janela excede determinado valor limite. Os finais dos segmentos são identificados quando a variação total do sinal no interior da janela é inferior a um segundo valor de *threshold*.

O quarto método (MTD4) é baseado no método de segmentação utilizado em (PATTICHIS *et al.* 1995). Os pontos de início do segmento são tais que, em uma janela à esquerda do ponto, o sinal mantém-se abaixo de determinado *threshold*. Os pontos de final de segmento, de forma similar, são tais que, em uma janela à direita do ponto, o sinal mantém-se abaixo do *threshold*.

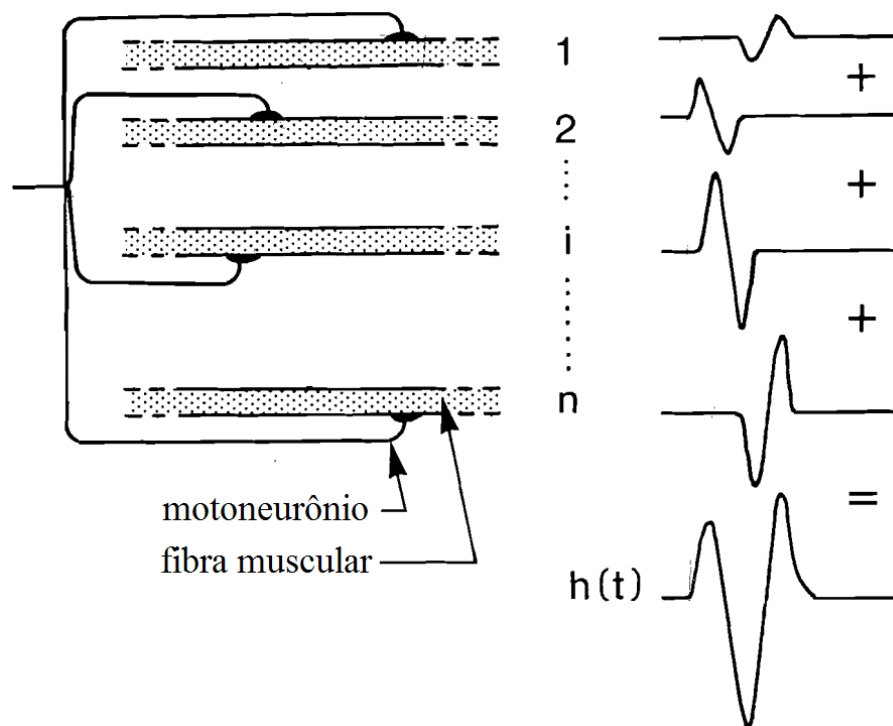
Utilizando valores de RMS, variância e frequência mediana dos sinais segmentados, uma rede neural artificial será treinada para classificar entre possíveis movimentos. O objetivo final deste estudo é avaliar a influência dos métodos de segmentação nas taxas de acerto de classificação.

2 Revisão Bibliográfica

2.1 Sinais de Eletromiografia

Sinais de EMG podem ser adquiridos por eletrodos posicionados na superfície da pele (eletrodo não invasivo) ou por agulhas introduzidas no tecido muscular (eletrodo invasivo). Sinais de EMG são compostos por potenciais de ação de fibras musculares organizadas em unidades funcionais chamadas de "unidades motoras" (MU - *Motor Unit*) (DE LUCA *et al.* 2006). Uma unidade motora é composta por um neurônio motor e as fibras musculares que ele inerva, e é a entidade fundamental que controla a ativação de músculos estriados (BUCHTAL and SCHMALBRUCH, 1980). A soma algébrica dos potenciais de ação de todas as fibras de uma unidade motora é chamada de "potencial de ação da unidade motora", ou em inglês, MUAP (*Motor Unit Action Potential*) (ALMEIDA, 1997). A Figura 1 apresenta a composição de uma MUAP a partir da soma dos potenciais das fibras de uma unidade motora.

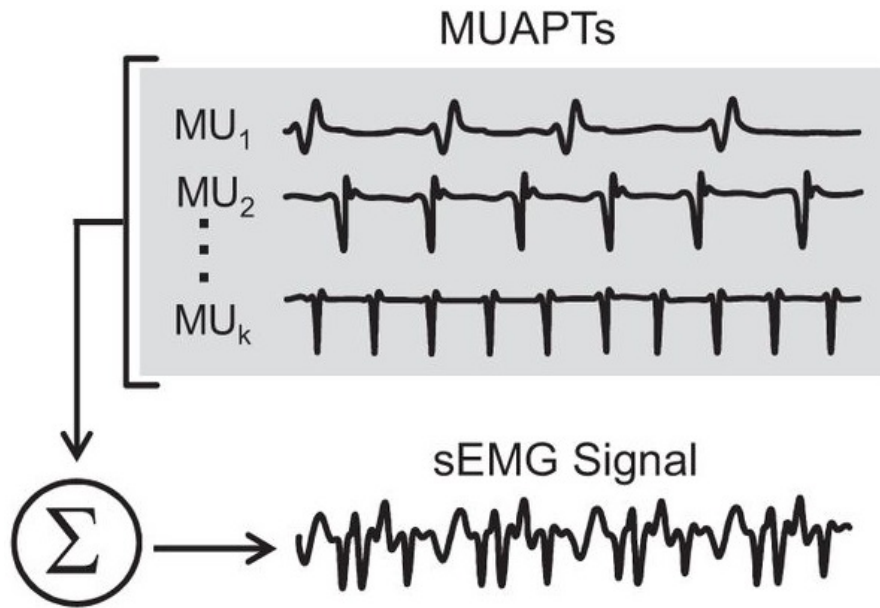
Figura 1: Soma de potenciais de ação das n fibras de uma unidade motora, formando uma MUAP $h(t)$.



Fonte: adaptado de BASMAJIAN & DE LUCA, 1985

Dependendo do método utilizado para aquisição de EMG, é comum a captura da contribuição de mais de uma unidade motora no mesmo canal. A influência de uma unidade motora no sinal adquirido depende principalmente da distância das fibras musculares ao ponto de aquisição (HAMMARBERG and STERNAD, 2002). Sinais de EMG de longa duração são constituídos por sequências temporais de MUAPs, também conhecidas como MUAPTs (*MUAP Trains*). A Figura 2 exemplifica MUAPTs de diferentes MUs que somam-se para formar um sinal de EMG de longa duração.

Figura 2: MUAPTs de diferentes MUs somam-se para compor o sinal adquirido por um canal de EMG.



Fonte: adaptado de KLINE & DE LUCA, 2014

2.2 Métodos de Segmentação

Esta seção descreve os métodos de segmentação que foram utilizados como base teórica para os métodos desenvolvidos neste trabalho. Nota-se que nomes utilizados para variáveis e constantes (por exemplo, sinal a ser segmentado ' x ', *threshold* ' T ', etc.) foram determinados pelo autor deste estudo, não necessariamente sendo estes utilizados nos métodos originais.

Para as definições dos métodos 3 e 4 (MTD3 e MTD4) são utilizados os termos BEP (*beginning extraction point*, ponto inicial de um segmento) e EEP (*ending extraction point*, ponto final de um segmento), também utilizados em (PATTICHIS *et al.* 1995).

2.2.1 Método 1 - método iterativo utilizando *thresholding* para detecção de centros de segmentos de comprimento constante (MTD1)

Este é o método iterativo de segmentação utilizado em (CHAUVET *et al.* 2001). As definições da Tabela 1 serão utilizados para descrever este método.

Tabela 1: Parâmetros utilizados para definir MTD1.

Nome	Descrição
x	Sinal a ser segmentado
L	Comprimento total do sinal a ser segmentado
l	Comprimento desejado para os segmentos
T_k	Valor de <i>threshold</i> para a iteração k
T_{lim}	Valor de limite inferior para o <i>threshold</i>
q	Fração de T_{k-1} para determinação de T_k
N_k	Número total de candidatos para centros de segmentos identificados na iteração k
r_k	Razão entre número de candidatos identificados na iteração k e o comprimento total do sinal
r_{target}	Razão mínima esperada para r_k , utilizada para determinar o final do método

Inicialmente, determina-se o valor de *threshold* T_0 equivalente ao máximo absoluto do sinal a ser segmentado x (Equação (2.1)). O valor T_k é atualizado em cada iteração k como sendo uma fração q de T_{k-1} (Equação (2.2)). No trabalho de (CHAUVET *et al.* 2001), este valor q foi empiricamente determinado em 90%.

$$T_0 = \max(x) \quad (2.1)$$

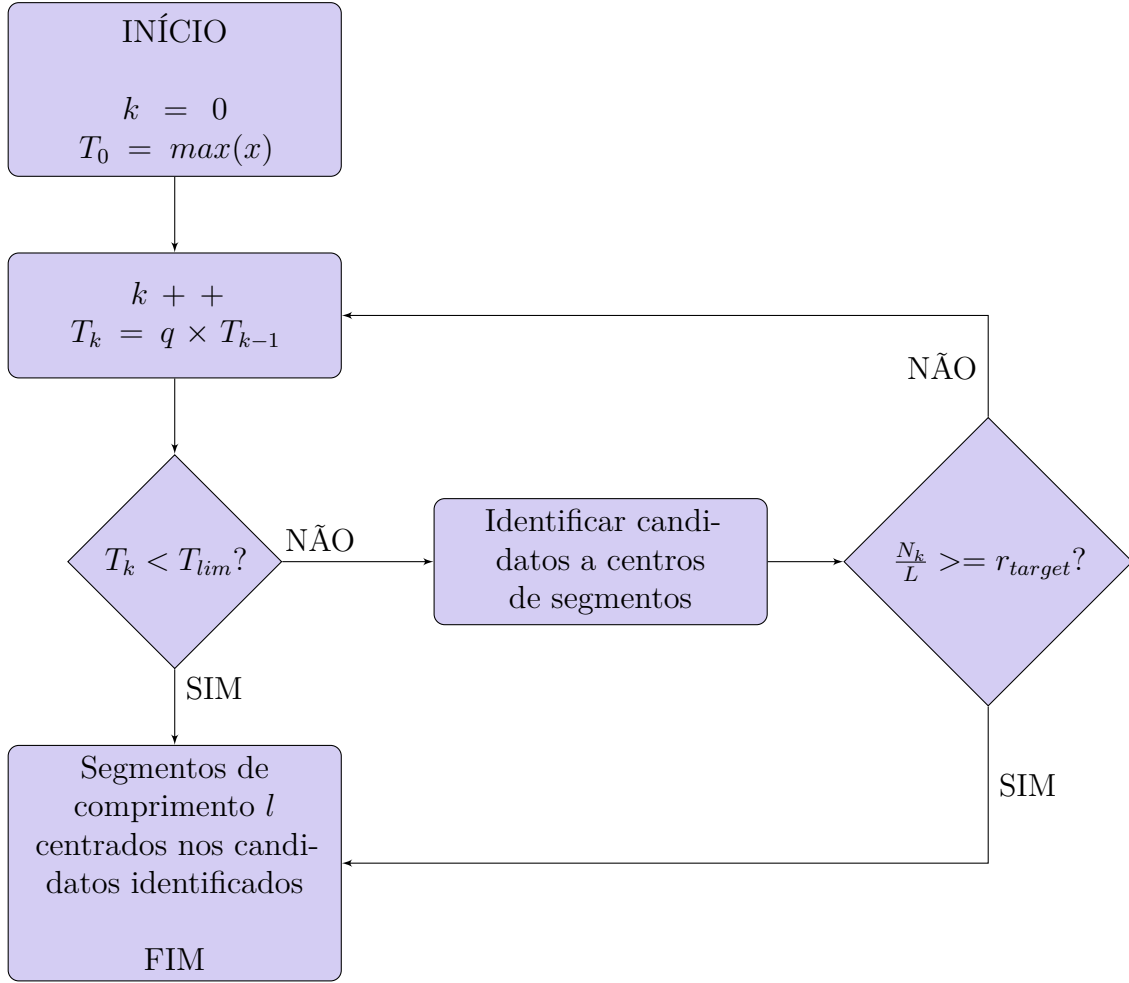
$$T_k = q \times T_{k-1} \quad (2.2)$$

Pontos do sinal acima do valor de T_k são possíveis candidatos para centros de segmentos. Caso exista mais de um possível candidato em uma vizinhança bilateral de l amostras do sinal, apenas o ponto de maior amplitude nesta vizinhança é considerado. Para determinar o final do método, avalia-se a razão r_k entre a quantidade identificada de candidatos N_k e o comprimento total do sinal L (Equação (2.3)). Caso r_k seja menor que um valor predeterminado r_{target} , calcula-se T_{k+1} para realização da próxima iteração (Equação (2.2)). Caso r_k seja maior ou igual ao valor predeterminado r_{target} , encerra-se o método e os segmentos são tomados como janelas de sinal de comprimento l , centradas nos candidatos identificados na última iteração.

$$r_k = \frac{N_k}{L} \quad (2.3)$$

Adicionalmente, o estabelecimento de um valor limite mínimo para *threshold* T_{lim} garante que o método não entre em laço infinito e evita detecção de segmentos em trechos de baixa atividade muscular. Caso o valor de *threshold* T_k para a iteração atual seja inferior a T_{lim} , encerra-se o processo iterativo. O método de segmentação MTD1 é representado pelo fluxograma da Figura 3.

Figura 3: Fluxograma representativo do MTD1.



2.2.2 Método 2 - método não iterativo utilizando *thresholding* para detecção de centros de segmentos de comprimento constante (MTD2)

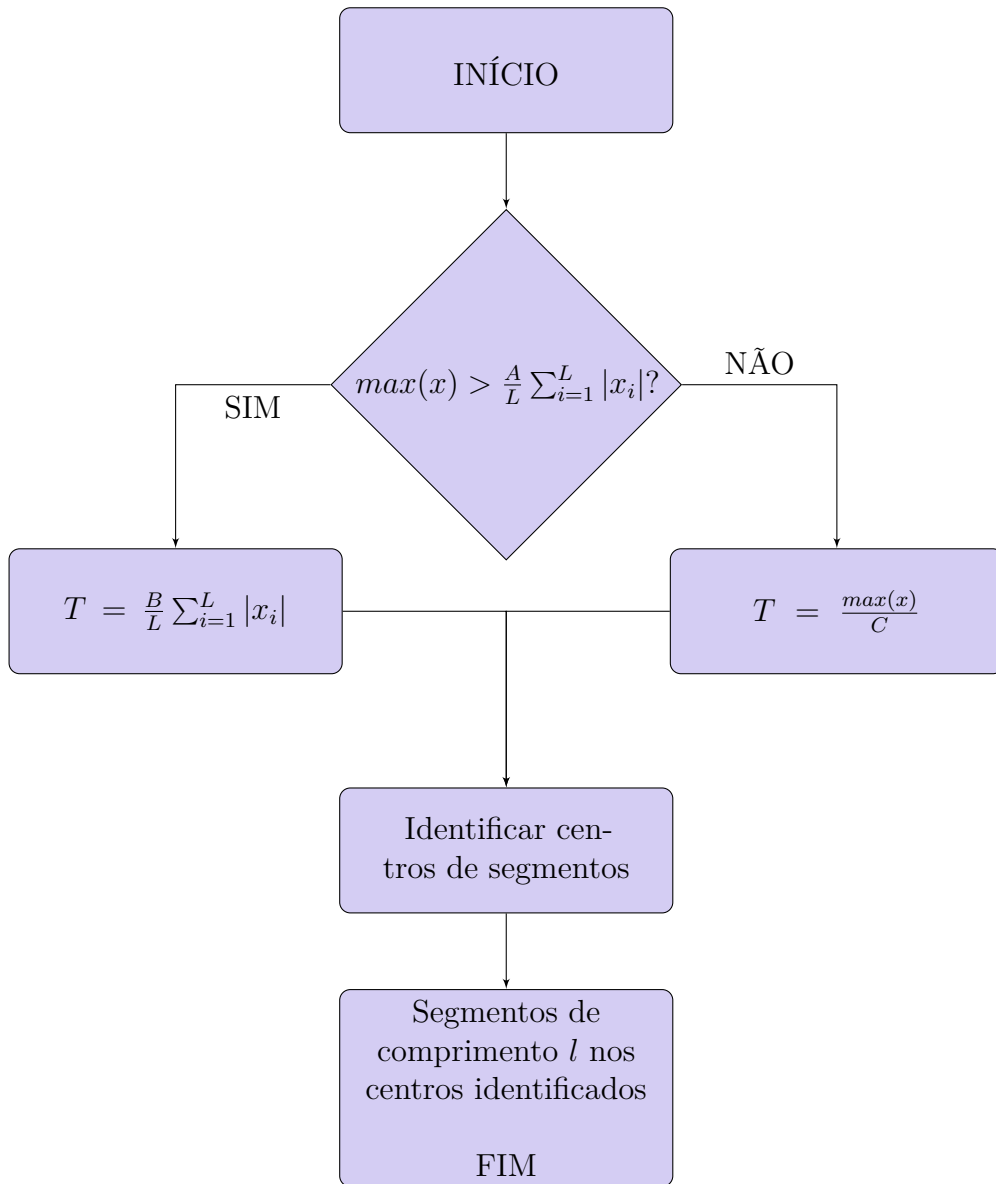
Este é o método de segmentação utilizado em (KATSIS *et al.* 2006), que será descrito pelas definições da Tabela 2. Primeiramente, seleciona-se entre dois métodos de cálculo de *threshold* T : ou utiliza-se T como múltiplo da média aritmética do sinal x ; ou T é uma fração do valor máximo do sinal x . O trabalho de (KATSIS *et al.* 2006) utiliza a relação do fluxograma da Figura 4 para o cálculo de *threshold* T .

De forma similar ao MTD1, os pontos do sinal que tiverem valor acima de T são considerados possíveis candidatos para centros de segmentos. Para os possíveis candidatos

Tabela 2: Parâmetros utilizados para definir MTD2.

Nome	Descrição
x	Sinal a ser segmentado
L	Comprimento total do sinal a ser segmentado
l	Comprimento desejado para os segmentos
T	Valor de <i>threshold</i>
A	Coefficiente utilizado para decisão de método de cálculo de T
B	Múltiplo da média aritmética do sinal x para obtenção de T
C	Fração do valor máximo do sinal x para cálculo de T

Figura 4: Fluxograma representativo do MTD2.



que estiverem afastados de uma distância inferior a l , apenas o candidato de maior amplitude é considerado. Em (KATSIS *et al.* 2006) utilizou-se coeficientes A , B e C respectivamente de 30, 5 e 5, com comprimento l de 121 amostras.

2.2.3 Método 3 - método com janela deslizante para detecção de BEP e EEP de segmentos utilizando variação total (MTD3)

Este é o método de segmentação utilizado em (GUT and MOSCHYTZ, 2000). As definições da Tabela 3 serão utilizados para descrever este método.

Tabela 3: Parâmetros utilizados para definir MTD3.

Nome	Descrição
x	Sinal a ser segmentado
W	Comprimento da janela deslizante utilizada pelo método
w_0	Número da amostra mais a esquerda da janela. Determina a posição instantânea da janela
β	Declividade média do sinal x contido na janela deslizante
B	Valor limite para declividade média que determina um BEP
γ	Variação total do sinal x contido na janela deslizante
C	Valor limite para variação total que determina um EEP

Uma janela deslizante de comprimento W percorre o sinal da esquerda para a direita. Caso a declividade média β do trecho de sinal contido pela janela, calculado pela Equação (2.4), exceda um limite B , o ponto mais à esquerda da janela w_0 determina a BEP de um segmento.

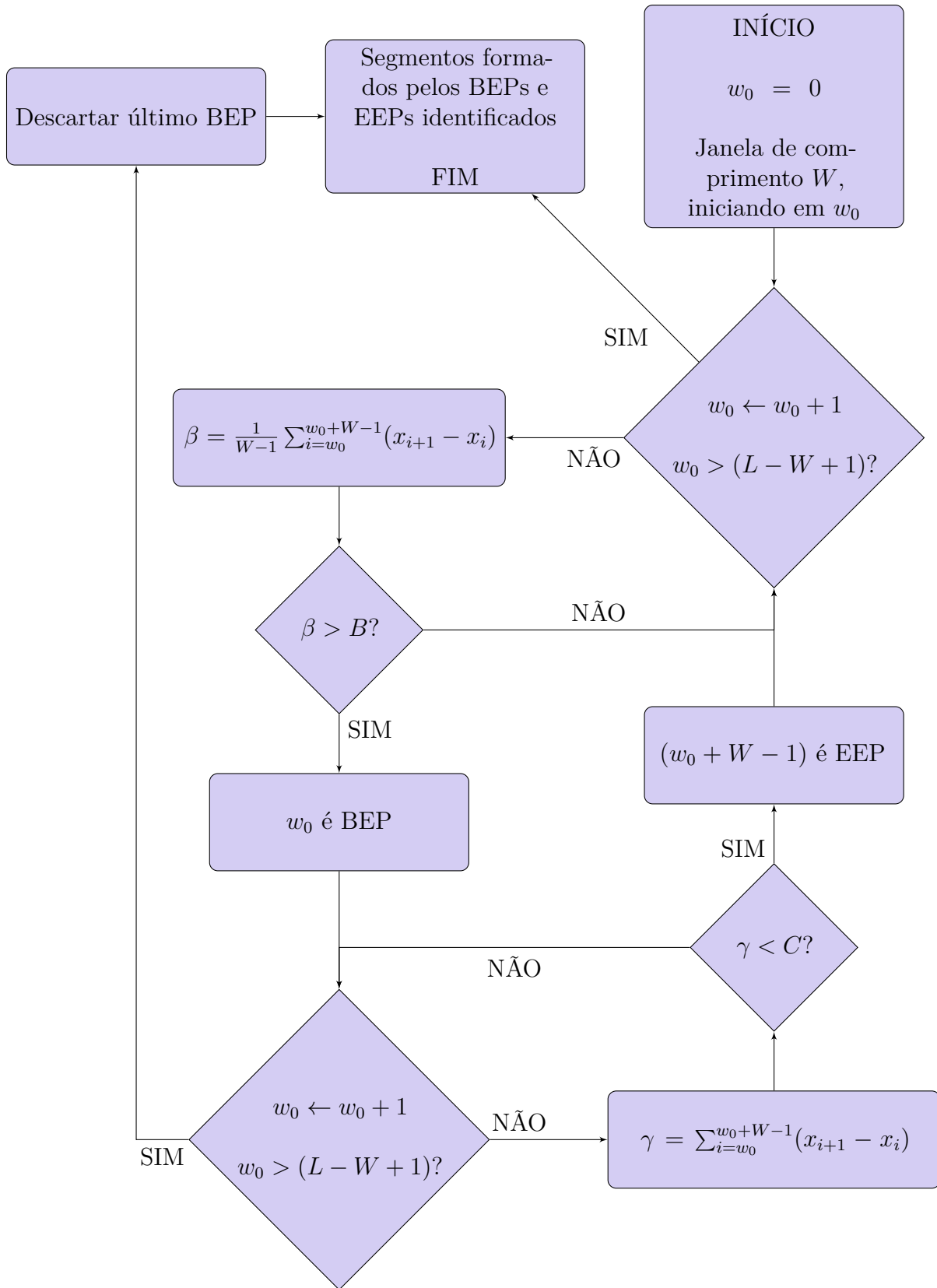
$$\beta = \frac{1}{W} \sum_{i=w_0+1}^W (x_i - x_{i-1}) \quad (2.4)$$

O EEP do correspondente segmento é então obtido como o ponto mais à direita da janela ($w_0 + W - 1$) quando a variação total γ , dado pela Equação (2.5), do trecho de sinal contido pela janela for menor que um limite C .

$$\gamma = \sum_{i=w_0+1}^W (x_i - x_{i-1}) \quad (2.5)$$

O MTD3 pode ser representado pelo fluxograma da Figura 5.

Figura 5: Fluxograma representativo do MTD3.



2.2.4 Método 4 - método com janela deslizante para detecção de BEP e EEP de segmentos utilizando *thresholding* (MTD4)

Este é o método de segmentação utilizado em (PACHITTIS *et al.* 1995). As definições da Tabela 4 serão utilizados para descrever este método.

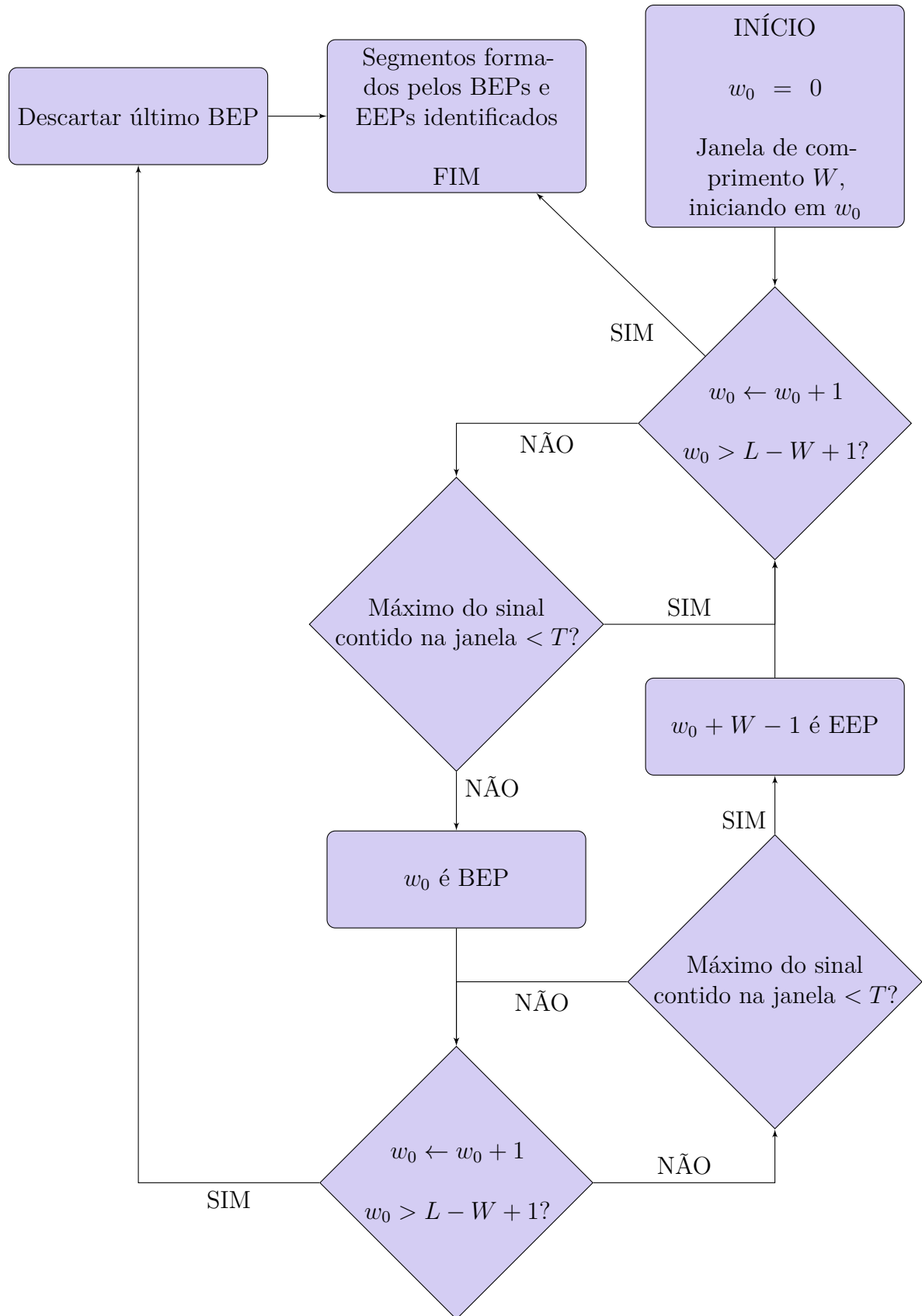
Tabela 4: Parâmetros utilizados para definir MTD4.

Nome	Descrição
x	Sinal a ser segmentado
L	Comprimento total do sinal x
W	Comprimento da janela deslizante utilizada pelo método
T	Valor de <i>threshold</i>

Uma janela deslizante de comprimento W com início em w_0 percorre o sinal da esquerda para a direita. Os BEPs dos segmentos são pontos w_0 tais que o valor máximo do sinal contido pela janela permanece abaixo do valor de *threshold* T . Para sequências de pontos consecutivos que atendam esta especificação, seleciona-se o último ponto (ponto mais à direita). As EEPs são identificadas de forma similar, sendo os primeiros pontos $w_0 + W - 1$ após as BEPs nos quais o sinal contido pela janela permanece abaixo do valor de *threshold* T .

No trabalho de (PACHITTIS *et al.* 1995), utilizou-se janelas de comprimento W correspondente a 3 *ms* de aquisição e *threshold* T de $\pm 40 \mu V$ (o sinal de EMG segmentado não era retificado). O fluxograma da Figura 6 representa o MTD4.

Figura 6: Fluxograma representativo do MTD4.



2.3 Base de Dados Utilizadas

2.3.1 Base de Dados Ninapro

2.3.2 Base de Dados do IEE

3 Metodologia Experimental

3.1 Preprocessamento

Esta seção descreve as operações de preprocessamento aplicadas aos sinais de eletromiografia anterior à implementação dos métodos de segmentação.

3.1.1 Retificação de Sinal

O sinal, adquirido em

3.1.2 Normalização

3.1.3 MTD1 e MTD2: uso da função *findpeaks* para detecção de centros de segmentos

3.1.4 Obtenção de valores para parâmetros dos métodos

3.2 Rede Neural Artificial

3.2.1 Características dos segmentos utilizadas pela rede

3.2.2 Treinamento, Validação e Teste

4 Capítulo temporário: orientação

Caro professor Balbinot,

Percebes que não fiz avanços significativos no texto, pois preocupei-me em trabalhar nas implementações dos métodos. Seguem o próximos passos que pretendo seguir para finalizar este trabalho.

- Falar com Vinicius para obter o cálculo de frequência mediana que ele está utilizando.
- Falar com Fernanda sobre o processo de aquisição da database do laboratório.
- Obtenção de parâmetros para os métodos.
- Implementar a rede neural e analisar resultados.

Sobre o préprocessamento: primeiramente, retifico os canais do sinal (utilizando $abs()$). Para lidar com a existência de múltiplos canais do sinal de EMG, realizo a soma de todos os canais, obtendo assim um único sinal x_{sum} . Posteriormente, filtro este sinal x_{sum} com um FIR com frequência de corte 20 Hz. Obtenho as posições de segmentos do sinal original (anterior à soma) aplicando os métodos (MTD1-4) a este sinal x_{sum} filtrado. Estás de acordo com os passos realizado?

Estou receoso sobre como justificar o uso de um FIR nesta baixa frequência, visto que sinais de EMG têm banda de até 500 Hz. Porém, a forma do sinal obtido pela soma dos canais é relacionada aos movimentos realizados pela pessoa, que têm esta frequência menor associada. Além disto, para os métodos de segmentação, a ocorrência de *spikes* no sinal afeta negativamente os resultados dos métodos, e a filtragem em 20 Hz atenua tais ocorrências. Consideras, portanto, devidamente justificável o uso de FIR em 20 Hz?

Ainda estou decidindo a melhor estratégia para determinar os parâmetros dos métodos (de formas que não sejam completamente empíricas). Tenho em mente implementar os métodos realizando “*sweeping*” dos parâmetros, utilizando número de segmentos obtidos e distâncias entre eles como métrica da determinação dos melhores parâmetros. Tens alguma sugestão a dar a respeito desta ou outra estratégia?

Sobre trabalhos futuros, gostaria de sugerir a adaptação dos métodos, que são de natureza essencialmente não-causal (*offline*), para métodos causais (*online*). De acordo?

Grato, Vicente

5 Resultados e Discussões

6 Conclusões

7 Propostas de Futuros Trabalhos

Apêndices

APÊNDICE A – Função em Matlab para MTD1

```

%%
%   MTD1 - metodo iterativo utilizando thresholding para deteccao de
%   centros de segmentos de comprimento constante
%
% Argumentos: (para mais detalhes, refira a descricao do MTD1)
%   x - matriz cujas colunas sao canais do sinal a ser segmentado
%   l - comprimento desejado para os segmentos
%       (deve ser inteiro maior que zero)
%   q - razao de atualizacao entre iteracoes para valor de threshold
%       (deve ser entre 0 e 1)
%   r_target - razao minima esperada entre numero de segmentos e comprimento
%              total de sinal
%              (deve ser maior que zero)
%   T_lim - fracao do maximo do sinal para limite inferior de threshold
%           (deve ser entre 0 e 1)
%
% Retorno:
%   x_seg - cell array com canais segmentados
%   centerLocs - posicoes centrais dos segmentos
%%

function [x_seg, centerLocs] = seg_mtd1(x, l, q, r_target, T_lim)
%% Preprocessamento

% Obtem comprimento do sinal e numero de canais
[L, numberOfChannels] = size(x);

% Retificacao
x_ret = abs(x);

% Normalizacao
x_norm = zeros(L, numberOfChannels);
for currentChannel = 1:numberOfChannels
    x_norm(:,currentChannel) = ...
        x_ret(:,currentChannel)./max(x_ret(:,currentChannel));
end

%% Metodo

```

```

% Cell array para armazenar posicoes dos segmentos identificados
centerLocsCell = cell(1,numberOfChannels);

for currentChannel = 1:numberOfChannels

    % Processo iterativo
    T_k = 1; % canais normalizados, seus valores maximos sempre sao 1
    targetReached = false;
    while ~targetReached

        T_k = q*T_k; % calcula threshold desta iteracao
        if T_k < T_lim % verifica se o limite de valor de threshold foi atingido
            break
        end

        % Identifica candidatos a centros de segmentos
        [~, centerLocsCell{1,currentChannel}] = ...
            findpeaks(double(x_norm(:,currentChannel)), ...
                'MinPeakHeight', T_k, 'MinPeakDistance',1);

        % Determina o encerramento do processo iterativo
        targetReached = (length(centerLocsCell{1,currentChannel})/L >= r_target);
    end

end

%% Segmentacao do sinal

% Maximo numero de segmentos detectados
numberOfSegments = 0;
for currentChannel = 1:numberOfChannels
    currentChannelNumberOfSegments = length(centerLocsCell{1,currentChannel});
    if currentChannelNumberOfSegments > numberOfSegments
        numberOfSegments = currentChannelNumberOfSegments;
    end
end

% Clustering dos centros de segmentos detectados
centerLocsArray = cell2mat(centerLocsCell');
[~,C] = kmeans(centerLocsArray,numberOfSegments);
centerLocs = sort(round(C));

% Segmentacao
x_seg = cell(numberOfSegments,numberOfChannels);
for currentChannel = 1:numberOfChannels
    for currentSegment = 1:numberOfSegments
        if mod(1,2) == 0 % se 1 for par

```

```

    if (centerLocs(currentSegment)-l/2) < 1 % segmento muito a esquerda
        x_seg{currentSegment,currentChannel} = ...
            x(1:centerLocs(currentSegment)+l/2 - 1, currentChannel);
    else if (centerLocs(currentSegment)+l/2 - 1) > L % segmento muito a direita
        x_seg{currentSegment,currentChannel} = ...
            x(centerLocs(currentSegment)-l/2:L, currentChannel);
    else
        x_seg{currentSegment,currentChannel} = ...
            x(centerLocs(currentSegment)-l/2: ...
                centerLocs(currentSegment)+l/2 - 1, currentChannel);
    end
end
else % se l for impar
    if (centerLocs(currentSegment)-(l+1)/2) < 1 % segmento muito a esquerda
        x_seg{currentSegment,currentChannel} = ...
            x(1:centerLocs(currentSegment)+l/2 - 1, currentChannel);
    else if (centerLocs(currentSegment)+(l+1)/2 - 1) > L % segmento muito a direita
        x_seg{currentSegment,currentChannel} = ...
            x(centerLocs(currentSegment)-l/2:L, currentChannel);
    else
        x_seg{currentSegment,currentChannel} = ...
            x(centerLocs(currentSegment)-(l+1)/2: ...
                centerLocs(currentSegment)+(l+1)/2 - 1, currentChannel);
    end
end
end
end
end
end
end

```

APÊNDICE B – Função em Matlab para MTD2

```

%%
%   MTD2 - metodo nao iterativo utilizando thresholding para deteccao de
%   centros de segmentos de comprimento constante
%
% Argumentos: (para mais detalhes, refira a descricao do MTD2)
%   x - matriz cujas colunas sao canais do sinal a ser segmentado
%   l - comprimento desejado para os segmentos
%       (deve ser inteiro maior que zero)
%   A - coeficiente utilizado para decisao de metodo de calculo de threshold
%       (deve ser maior ou igual a 1)
%   B - multiplo da media aritmetica do sinal x para obtencao de threshold
%       (deve ser maior ou igual a 1)
%   C - fracao do valor maximo do sinal x para calculo de threshold
%       (deve ser maior ou igual a 1)
%
% Retorno:
%   x_seg - cell array com os canais segmentados
%   centerLocs - posicoes centrais dos segmentos
%%

function [x_seg, centerLocs] = seg_mtd2(x, l, A, B, C)

%% Preprocessamento

% Obtem comprimento do sinal e numero de canais
[L, numberOfChannels] = size(x);

% Retificacao de sinal
x_ret = abs(x);

% Suavizacao utilizando media movel
x_smooth = reshape(smooth(x_ret, 32), L, numberOfChannels);

%% Metodo

% Cell array para armazenar posicoes dos segmentos identificados
centerLocsCell = cell(1,numberOfChannels);

for currentChannel = 1:numberOfChannels

```

```

    % Calculo do threshold
    maxValue = max(x_smooth(:,currentChannel));
    meanValue = mean(x_smooth(:,currentChannel));
    if maxValue > (A*meanValue)
        T = B*meanValue;
    else
        T = maxValue/C;
    end

    % Identifica centros de segmentos
    [~, centerLocsCell{1,currentChannel}] = ...
        findpeaks(double(x_smooth(:,currentChannel)), ...
            'MinPeakHeight', T, 'MinPeakDistance',1);
end

%% Segmentacao dos canais

% Maximo numero de segmentos detectados
numberOfSegments = 0;
for currentChannel = 1:numberOfChannels
    currentChannelNumberOfSegments = length(centerLocsCell{1,currentChannel});
    if currentChannelNumberOfSegments > numberOfSegments
        numberOfSegments = currentChannelNumberOfSegments;
    end
end

% Clustering dos centros de segmentos detectados
centerLocsArray = cell2mat(centerLocsCell');
idx = kmeans(centerLocsArray,numberOfSegments);

% Media dos clusters
centerLocsMean = zeros(numberOfSegments,1);
for currentCluster = 1:numberOfSegments
    centerLocsMean(currentCluster) = ...
        mean(centerLocsArray(idx == currentCluster));
end
centerLocs = sort(round(centerLocsMean));

% Segmentacao
x_seg = cell(numberOfSegments,numberOfChannels);
for currentChannel = 1:numberOfChannels
    for currentSegment = 1:numberOfSegments
        if mod(1,2) == 0 % se 1 for par
            x_seg{currentSegment,currentChannel} = ...
                x(centerLocs(currentSegment)-1/2: ...
                    centerLocs(currentSegment)+1/2 - 1, currentChannel);
        end
    end
end

```

```
        else % se l for impar
            x_seg{currentSegment,currentChannel} = ...
                x(centerLocs(currentSegment)-(l+1)/2: ...
                    centerLocs(currentSegment)+(l+1)/2 - 1, currentChannel);
        end
    end
end
end
end
```

APÊNDICE C – Função em Matlab para MTD3

```

%%
%   MTD3 - metodo com janela deslizando para deteccao de BEP e EEP de segmentos
%   utilizando variacao total
%
% Argumentos: (para mais detalhes, refira a descricao do MTD3)
%   x - matriz cujas colunas sao canais do sinal a ser segmentado
%   W - comprimento da janela deslizando utilizada pelo metodo
%       (deve ser inteiro maior que zero)
%   B - valor limite para declividade media que determina um BEP
%       (deve ser maior que zero)
%   C - valor limite para variacao total que determina um EEP
%       (deve ser maior que zero)
%
% Retorno:
%   x_seg - cell array com os canais segmentados
%   centerLocs - posicoes centrais dos segmentos
%%

function x_seg = seg_mtd3(x, B, C)

%% Preprocessamento

% Obtem comprimento do sinal e numero de canais
[L, numberOfChannels] = size(x);

% Retificacao de sinal
x_ret = abs(x);

% Suavizacao utilizando media movel
x_smooth = reshape(smooth(x_ret, 32), L, numberOfChannels);

%% Metodo

% Array logico para BEPs e EEPs detectados
BEPsFlags = false(L,1);
EEPsFlags = false(L,1);

% Indicador se a janela procura por BEP ou EEP
searchBEP = true;

```

```
% Janela deslizante
for w0 = 1:L-W
    if( (mean(diff(x_filt(w0:w0+W-1))) > B) && searchBEP ) % Deteccao de BEP
        BEPsFlags(w0) = 1;
        searchBEP = false;
    end
    if( (sum(diff(x_filt(w0:w0+W-1))) < C) && ~searchBEP ) % Deteccao de EEP
        EEPsFlags(w0+W-1) = 1;
        searchBEP = true;
    end
end

% Posicoes de EEPs e BEPs
BEPsLocs = find(BEPsFlags);
EEPsLocs = find(EEPsFlags);

% Caso tenha sido detectada uma BEP sem respectivo EEP, elimina ultimo BEP
numberOfSegments = length(BEPsLocs);
if( numBEPs > length(EEPsLocs) )
    BEPsLocs(end) = [];
end

% Segmentacao dos canais
x_seg = cell(numberOfSegments,numberOfChannels);
for currentChannel = 1:numberOfChannels
    for currentSegment = 1:numberOfSegments
        x_seg{currentSegment,currentChannel} = ...
            x(BEPsLocs(currentSegment):EEPsLocs(currentSegment));
    end
end
end
```


APÊNDICE D – Função em Matlab para MTD4

```

%%
%   MTD4 - metodo com janela deslizando para deteccao de BEP e EEP de segmentos
%   utilizando thresholding
%
% Argumentos: (para mais detalhes, refira a descricao do MTD4)
%   x - matriz cujas colunas sao canais do sinal a ser segmentado
%   W - comprimento da janela deslizando utilizada pelo metodo
%       (deve ser inteiro maior que zero)
%   T - valor de threshold
%       (deve ser maior que zero)
%
% Retorno:
%   x_seg - cell array com os canais segmentados
%   centerLocs - posicoes centrais dos segmentos
%%

function x_seg = seg_mtd4(x, W, T)

%% Preprocessamento

% Obtem comprimento do sinal e numero de canais
[L, numberOfChannels] = size(x);

% Retificacao de sinal
x_ret = abs(x);

% Soma dos canais
x_sum = zeros(L,1);
for currentChannel = 1:numberOfChannels
    x_sum = x_sum + x_ret(:,currentChannel);
end

% FIR passa-baixas em 20 Hz
x_filt = filter(fir1(255,0.01),1,x_sum);

%% Metodo

% Array logico para BEPs e EEPs detectados
BEPsFlags = false(L,1);

```

```

EEPsFlags = false(L,1);

% Indicador se a janela procura por BEP ou EEP
searchBEP = true;

% Janela deslizante
for w0 = 1:L-W
    if( ~(max(x_filt(w0:w0+W-1)) < T) && searchBEP ) % Deteccao de BEP
        BEPsFlags(w0) = 1;
        searchBEP = false;
    end
    if( (max(x_filt(w0:w0+W-1)) < T) && ~searchBEP ) % Deteccao de EEP
        EEPsFlags(w0+W-1) = 1;
        searchBEP = true;
    end
end

% Posicoes de EEPs e BEPs
BEPsLocs = find(BEPsFlags);
EEPsLocs = find(EEPsFlags);

% Caso tenha sido detectada uma BEP sem respectivo EEP, elimina ultimo BEP
numberOfSegments = length(BEPsLocs);
if( numBEPs > length(EEPsLocs))
    BEPsLocs(end) = [];
end

% Segmentacao dos canais
x_seg = cell(numberOfSegments,numberOfChannels);
for currentChannel = 1:numberOfChannels
    for currentSegment = 1:numberOfSegments
        x_seg{currentSegment,currentChannel} = ...
            x(BEPsLocs(currentSegment):EEPsLocs(currentSegment));
    end
end
end

```